VISUALIZATION METHOD FOR SOFTWARE ARCHITECTURE AND APPARATUS

[0001]

TECHNICAL FIELD

[0002] This application relates to the field of computer technologies, and in particular, to a visualization method for a software architecture and an apparatus.

BACKGROUND

[0003] One piece of software may be divided into a plurality of different functional components, and a software architecture is used to describe composition of the software and an association relationship between the functional components in the software. To more intuitively display a software architecture of the piece of software, currently, a two-dimensional view may be used to display each functional component in the software in a two-dimensional plane, and display the software architecture by using a shape and a connection line. For example, in the two-dimensional view, the shape may be used to represent a functional component, and different shapes represent different functional components. The connection line is used to represent an association relationship between the functional components.

[0004] However, the two-dimensional view is limited by dimensions, and a graph in a formed two-dimensional view cannot be rotated or translated. A method of displaying the software architecture in the two-dimensional view is not flexible and intuitive.

functional component 4 runs slowly.

[0080] A deployment view displays a deployment position of a functional component. A deployment position of the functional component in the two-dimensional view may also be displayed based on the two-dimensional logical view by using text information. As shown in FIG. 4, a deployment position of each functional component is displayed by using a text next to the functional component in FIG. 4. The text in FIG. 4 shows that the functional component 1, the functional component 2, and the functional component 3 are deployed in a server 1, the functional component 5 is deployed in a server 3, and the functional component 4 is deployed in a server 2.

[0081] In FIG. 2A to FIG. 4, the two-dimensional view is limited by dimensions, information that can be displayed is limited, and a display manner is not vivid and intuitive enough.

[0082] The following describes the three-dimensional view of the software architecture according to this embodiment of this application. Instances in the functional component are the same or different, and manners of displaying the software architecture by using the three-dimensional view are different. The following separately provides descriptions.

[0083] Manner 1: Instances in the functional component are different, and the instances in the functional component cooperate with each other to implement a function of the functional component.

[0084] In the three-dimensional view, a stereoscopic graph may be used to represent the functional component in the software architecture, and a connection line between the stereoscopic graphs may represent an association relationship between the functional components. An inclusion relationship between the stereoscopic graphs can represent an affiliation relationship between the functional components. An internal composition of the stereoscopic graph may represent the instance in the functional component.

[0085] FIG. 5A is a three-dimensional view of a software architecture according to an embodiment of this application. The three-dimensional view includes five functional components: a functional component 1, a functional component 2, a functional component 3, a functional component 4, and a functional component 5. The functional component 1 is represented by using a flat cuboid, and the functional component 2, the functional component 3, the functional component 4, and the functional component 5 are represented by using relatively high cuboids. Cuboids representing the functional component 2 and the functional component 3 are placed on the flat cuboid, indicating an affiliation relationship between the functional component 1 and the

stereoscopic graph (for example, a small cuboid) corresponding to a functional component included in the functional component is changed, and different running statuses of a functional component with a small granularity may be represented by using different colors.

[00113] FIG. 7 shows a three-dimensional deployment view according to an embodiment of this application. Based on the three-dimensional logical view shown in FIG. 5A, a three-dimensional graph that can indicate a deployment position of a functional component may be added, and the three-dimensional graph in which the functional component is located is used to indicate the position of the functional component.

[00114] In FIG. 7, a three-dimensional graph representing a server is added. The functional component 1, the functional component 2, and the functional component 3 are deployed in a server 1, the functional component 4 is deployed in a server 2, and the functional component 5 is deployed in a server 3. A deployment position of each functional component may be directly learned from FIG. 7. Compared with the two-dimensional deployment view shown in FIG. 4, the three-dimensional deployment view is easier to view.

[00115] The display module 130 may display the three-dimensional running view shown in FIG. 6 and the three-dimensional deployment view shown in FIG. 7 to the user. When viewing the three-dimensional running view or the three-dimensional deployment view, the user may also perform an operation such as rotation, translation, or scaling on the three-dimensional running view or the three-dimensional deployment view. In response to the operation of the user, the construction module 120 may rotate, translate, or scale a stereoscopic graph corresponding to a functional component in the three-dimensional running view or the three-dimensional deployment view. A manner in which the construction module 120 may rotate, translate, or scale the stereoscopic graph corresponding to the functional component in the three-dimensional running view or the three-dimensional deployment view is the same as a manner in which the construction module 120 may rotate, translate, or scale the stereoscopic graph corresponding to the functional component in the three-dimensional logical view. For details, refer to the foregoing content. Details are not described herein again.

[00116] It should be noted that the foregoing content describes the rotation, translation, and scaling operations. Embodiments of this application are not limited to the rotation, translation, and scaling operations. For example, the user may also trigger, by using a cursor or a gesture, the construction module 120 to delete a stereoscopic graph in the three-dimensional view (that is,

functional component in the three-dimensional running view or the three-dimensional deployment view. A manner in which the construction module 120 may rotate, move, or scale the stereoscopic graph corresponding to the functional component in the three-dimensional running view or the three-dimensional deployment view is the same as a manner in which the construction module 120 may rotate, move, or scale the stereoscopic graph corresponding to the functional component in the three-dimensional logical view. For details, refer to the foregoing content. Details are not described herein again.

[00147] It should be noted that, in addition to the logical view, the running view, and the deployment view, the three-dimensional view may further provide a mapping view of an entity apparatus and a functional component. When there is a correspondence between the functional component in the software architecture and the entity apparatus, it means that in actual application, a function of the functional component may be implemented by using the entity apparatus. The entity apparatus and the functional component having a correspondence may be drawn in the mapping view, and a connection between the entity apparatus and the functional component is used to represent the correspondence between the functional component and the entity apparatus. FIG. 11 shows a three-dimensional mapping view according to an embodiment of this application. A three-dimensional structure of a server chassis is drawn in the three-dimensional mapping view. There is a correspondence between the functional component 1, the functional component 2, and the functional component 3 and the server chassis. In FIG. 11, an arrow between the functional component and the server chassis is used to indicate the correspondence.

[00148] The foregoing content describes a method for displaying the software architecture by using the two-dimensional view and the three-dimensional view, and a method for changing the three-dimensional view. The following describes a method for constructing the two-dimensional view and the three-dimensional view. The method for constructing the two-dimensional view and the three-dimensional view includes two aspects: one is a method for preliminarily constructing a view of the software architecture, and the other is a method for updating the view of the software architecture. The method for updating the view of the software architecture may be classified into an update method based on update information, and a method for updating another view based on an updated view.

[00149] The following describes, with reference to FIG. 12, a method for preliminarily constructing a view of a software architecture according to an embodiment of this application. As

communicate with each other through the bus 1601.

[00231] The processor 1602 may include one or more general-purpose processors, for example, a central processing unit (central processing unit, CPU), or a combination of a CPU and a hardware chip. The hardware chip may be an application-specific integrated circuit (application-specific integrated circuit, ASIC), a programmable logic device (programmable logic device, PLD), or a combination thereof. The PLD may be a complex programmable logic device (complex programmable logic device, CPLD), a field-programmable gate array (field-programmable gate array, FPGA), generic array logic (generic array logic, GAL), or any combination thereof.

[00232] The memory 1604 may include a volatile memory (volatile memory), for example, a random access memory (random access memory, RAM). The memory 1604 may further include a non-volatile memory (non-volatile memory, NVM), for example, a read-only memory (read-only memory, ROM), a flash memory, a hard disk drive (hard disk drive, HDD), or a solid-state drive (solid-state drive, SSD). The memory 1604 may further include a combination of the foregoing types.

[00233] The memory 1604 stores executable code. The processor 1602 may read the executable code in the memory 1604 to implement a function, and may further communicate with another computing device through the communication interface 1603. The processor 1602 may further trigger the display 1605 to display information to a user, for example, information in FIG. 2A to FIG. 11 in embodiments of this application. In this embodiment of this application, the processor 1602 may implement functions of one or more modules (for example, one or more of the receiving module 110, the construction module 120, and the display module 130) of the display apparatus 100. In this case, the memory 1604 stores one or more modules (for example, one or more of the receiving module 110, the construction module 120, and the display module 130) of the display apparatus 100.

[00234] In this embodiment of this application, the processor 1602 in the plurality of computing devices 1600 may work in coordination, to perform a software code change method that supports a plurality of changes and that is provided in embodiments of this application.

[00235] FIG. 17 shows a system architecture according to an embodiment of this application. The system architecture includes a client 200 and a cloud device 300 on which a change apparatus is deployed. The client 200 is connected to the cloud device 300 by using a network. The cloud device 300 is located in a cloud environment, and may be a server or a virtual machine deployed in a cloud data center. FIG. 17 shows only an example in which the change apparatus is deployed on one cloud device 300. In a possible implementation, the change apparatus may be deployed on

CLAIMS

What is claimed is:

1. A visualization method for a software architecture, wherein the method comprises:

constructing a three-dimensional view based on received architecture information of software, and displaying the three-dimensional view, wherein the architecture information of the software indicates that the software comprises at least one functional component, and a stereoscopic graph in the three-dimensional view represents a functional component;

adjusting the three-dimensional view in response to an operation of a user on the three-dimensional view; and

displaying an adjusted three-dimensional view.

2. The method according to claim 1, wherein the adjusting the three-dimensional view in response to an operation of a user on the three-dimensional view comprises:

updating the architecture information of the software in response to the operation of the user on the three-dimensional view; and

adjusting the three-dimensional view based on updated architecture information of the software.

3. The method according to claim 1 or 2, wherein the architecture information of the software comprises at least one of the following: a name of the functional component in the software, an association relationship between functional components in the software, a computing resource occupied by the functional component, and an affiliation relationship between the functional components in the software.

4. The method according to any one of claims 1 to 3, wherein the displayed three-dimensional view indicates at least one of the following:

a connection line between stereoscopic graphs in the three-dimensional view is used to indicate the association relationship between functional components in the software, the stereoscopic graph in the three-dimensional view indicates the computing resource occupied by the functional component in the software, and an inclusion relationship between the stereoscopic graphs in the three-dimensional view is used to indicate the affiliation relationship between the functional components in the software.

5. The method according to any one of claims 1 to 4, wherein the stereoscopic graph in the

three-dimensional view indicates the computing resource occupied by the functional component in the software, and the adjusting the three-dimensional view in response to an operation of a user on the three-dimensional view comprises:

scaling the stereoscopic graph in response to a scaling operation of the user on the stereoscopic graph in the three-dimensional view, to adjust the computing resource occupied by the functional component corresponding to the stereoscopic graph.

6. The method according to claim 5, wherein an edge of the stereoscopic graph is used to indicate the computing resource occupied by the functional component, and the scaling the stereoscopic graph in response to a scaling operation of the user on the stereoscopic graph in the three-dimensional view comprises:

stretching the edge in response to a stretching operation of the user on the edge of the stereoscopic graph in the three-dimensional view, to adjust the computing resource corresponding to the edge occupied by the functional component corresponding to the stereoscopic graph; or

in response to a stretching operation of the user on a vertex of the stereoscopic graph in the three-dimensional view, proportionally scaling all edges of the stereoscopic graph in a direction of the stretching operation, to adjust a computing resource corresponding to each edge occupied by the functional component corresponding to the stereoscopic graph.

7. The method according to any one of claims 1 to 6, wherein the stereoscopic graph in the three-dimensional view comprises a three-dimensional graph, and a three-dimensional graph comprised in a stereoscopic graph indicates an instance in a functional component corresponding to the stereoscopic graph; and the adjusting the three-dimensional view in response to an operation of a user on the three-dimensional view comprises:

scaling the three-dimensional graph in response to a scaling operation of the user on the three-dimensional graph in the three-dimensional view, to adjust a computing resource occupied by the instance indicated by the three-dimensional graph.

8. The method according to any one of claims 1 to 7, wherein after the adjusting the three-dimensional view, the method further comprises:

updating a two-dimensional view based on the adjusted three-dimensional view, wherein the two-dimensional view is pre-constructed and is used to display the software architecture, and a two-dimensional graph in the two-dimensional view represents the functional component; and

displaying an updated two-dimensional view.

9. A display apparatus, wherein the apparatus comprises:

a receiving module, configured to receive architecture information of software, wherein the architecture information of the software indicates that the software comprises at least one functional component;

a construction module, configured to construct a three-dimensional view based on the architecture information of the software, wherein a stereoscopic graph in the three-dimensional view represents a functional component; and

a display module, configured to display the three-dimensional view, wherein

the construction module is further configured to adjust the three-dimensional view in response to an operation of a user on the three-dimensional view, and

the display module is further configured to display an adjusted three-dimensional view.

10. The apparatus according to claim 9, wherein when adjusting the three-dimensional view in response to the operation of the user on the three-dimensional view, the construction module is specifically configured to:

update the architecture information of the software in response to the operation of the user on the three-dimensional view; and

adjust the three-dimensional view based on updated architecture information of the software.

11. The apparatus according to claim 9 or 10, wherein the architecture information of the software comprises at least one of the following: a name of the functional component in the software, an association relationship between functional components in the software, a computing resource occupied by the functional component, and an affiliation relationship between the functional components in the software.

12. The apparatus according to any one of claims 9 to 11, wherein the displayed three-dimensional view indicates at least one of the following:

a connection line between stereoscopic graphs in the three-dimensional view is used to indicate the association relationship between functional components in the software, the stereoscopic graph in the three-dimensional view indicates the computing resource occupied by the functional component in the software, and an inclusion relationship between the stereoscopic graphs in the three-dimensional view is used to indicate the affiliation relationship between the functional components in the software.

13. The apparatus according to any one of claims 9 to 12, wherein the stereoscopic graph in

the three-dimensional view indicates the computing resource occupied by the functional component in the software, and when adjusting the three-dimensional view in response to the operation of the user on the three-dimensional view, the construction module is specifically configured to:

scale the stereoscopic graph in response to a scaling operation of the user on the stereoscopic graph in the three-dimensional view, to adjust the computing resource occupied by the functional component corresponding to the stereoscopic graph.

14. The apparatus according to claim 13, wherein an edge of the stereoscopic graph is used to indicate the computing resource occupied by the functional component, and when scaling the stereoscopic graph in response to the scaling operation of the user on the stereoscopic graph in the three-dimensional view, the construction module is specifically configured to:

stretch the edge in response to a stretching operation of the user on the edge of the stereoscopic graph in the three-dimensional view, to adjust the computing resource corresponding to the edge occupied by the functional component corresponding to the stereoscopic graph; or

in response to a stretching operation of the user on a vertex of the stereoscopic graph in the three-dimensional view, proportionally scale all edges of the stereoscopic graph in a direction of the stretching operation, to adjust a computing resource corresponding to each edge occupied by the functional component corresponding to the stereoscopic graph.

15. The apparatus according to any one of claims 9 to 14, wherein the stereoscopic graph in the three-dimensional view comprises a three-dimensional graph, and a three-dimensional graph comprised in a stereoscopic graph indicates an instance in a functional component corresponding to the stereoscopic graph; and when adjusting the three-dimensional view in response to the operation of the user on the three-dimensional view, the construction module is specifically configured to:

scale the three-dimensional graph in response to a scaling operation of the user on the three-dimensional graph in the three-dimensional view, to adjust a computing resource occupied by the instance indicated by the three-dimensional graph.

16. The apparatus according to any one of claims 9 to 15, wherein after adjusting the three-dimensional view, the construction module is further configured to:

update a two-dimensional view based on the adjusted three-dimensional view, wherein the two-dimensional view is pre-constructed and is used to display the software architecture, and a

two-dimensional graph in the two-dimensional view represents the functional component; and

display an updated two-dimensional view.

17. A computing device, wherein the computing device comprises a processor and a memory, wherein

the memory is configured to store computer program instructions; and

the processor invokes the computer program instructions in the memory to perform the method according to any one of claims 1 to 8.

18. A computing device cluster, wherein the computing device cluster comprises a plurality of computing devices, each computing device comprises a processor and a memory, and a memory in at least one computing device is configured to store computer program instructions; and

a processor in the at least one computing device invokes the computer program instructions stored in the memory to perform the method according to any one of claims 1 to 8.